The eXtreme Adaptive DSP Solution to Sensor Data Processing

Martin Vorbach
PACT XPP Technologies

Leo Mirkin
Sky Computers, Inc.

Abstract

The new ISR mobile autonomous sensor platforms present a difficult challenge to the current generation of high-speed DSP solutions. The high-bandwidth data streams must be processed with minimal power and heat dissipation. The programmatic needs on the new ISR platforms include a rapid pace of technology upgrades, a quick development cycle, and lower costs; especially reduction in the software development costs and risks. These goals conflict with high-cost and risk of ASIC design, high-cost of FPGA development, and power-hungry but easy to program general-purpose CPU and DSP.

The eXtreme architecture is a new class of programmable DSP devices that builds on decades of experience with the array processors, general-purpose CPU, DSP, and FPGA. It also takes into account economic realities of the semiconductor industry affecting product acceptance and longevity. The eXtreme architecture combines the right mixture of arithmetic processing elements, high-bandwidth I/O ports and embedded memories components for the data-flow intensive processing in a system-on-the-chip (SoC). It is all tied together by a data packet-switching auto synchronizing interconnect, and supported by the dynamic self-reconfiguration infrastructure to form a new class of DSP programmable devices.

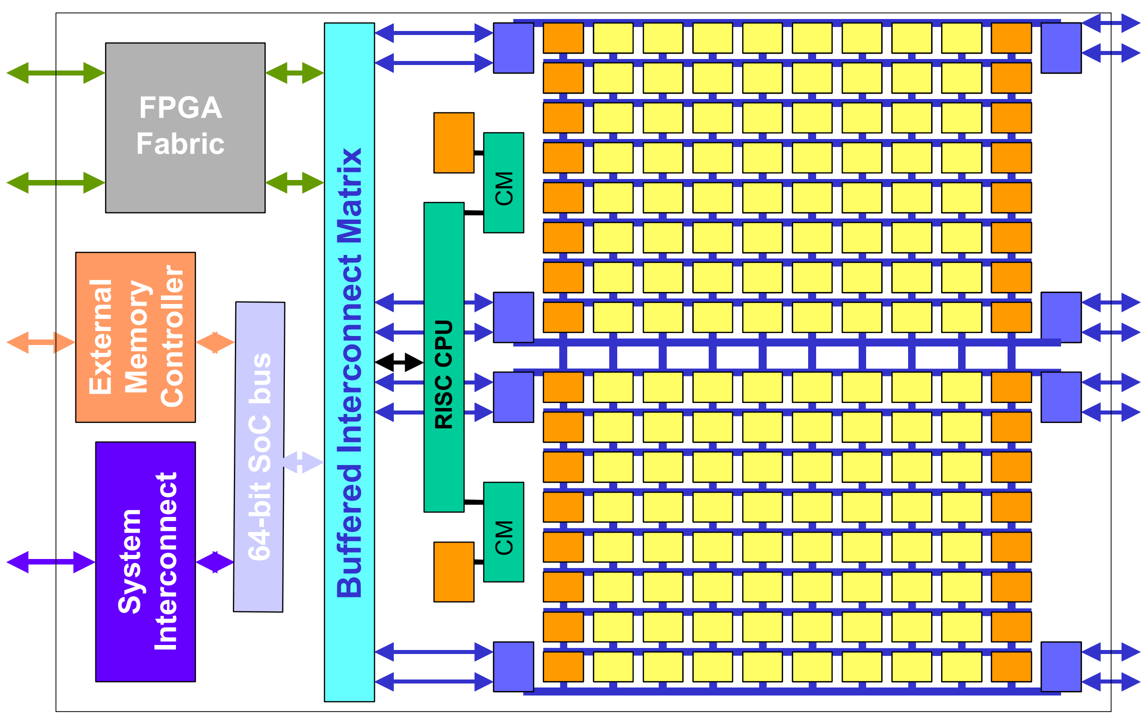


Figure 1. Example of eXtreme DSP - QuickMIPS-XPP

The purpose of the eXtreme architecture is to marry the application-specific computational efficiency of an ASIC with the programmability and familiar tools of a general-purpose CPU. The internal RISC processor and the packet-switching data interconnect enable a full range of programming options, including support for C programming language. It even leaves enough

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **20 AUG 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **The eXtreme Adaptive DSP Solution to Sensor Data Processing**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **PACT XPP Technologies; Sky Computers, Inc.**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

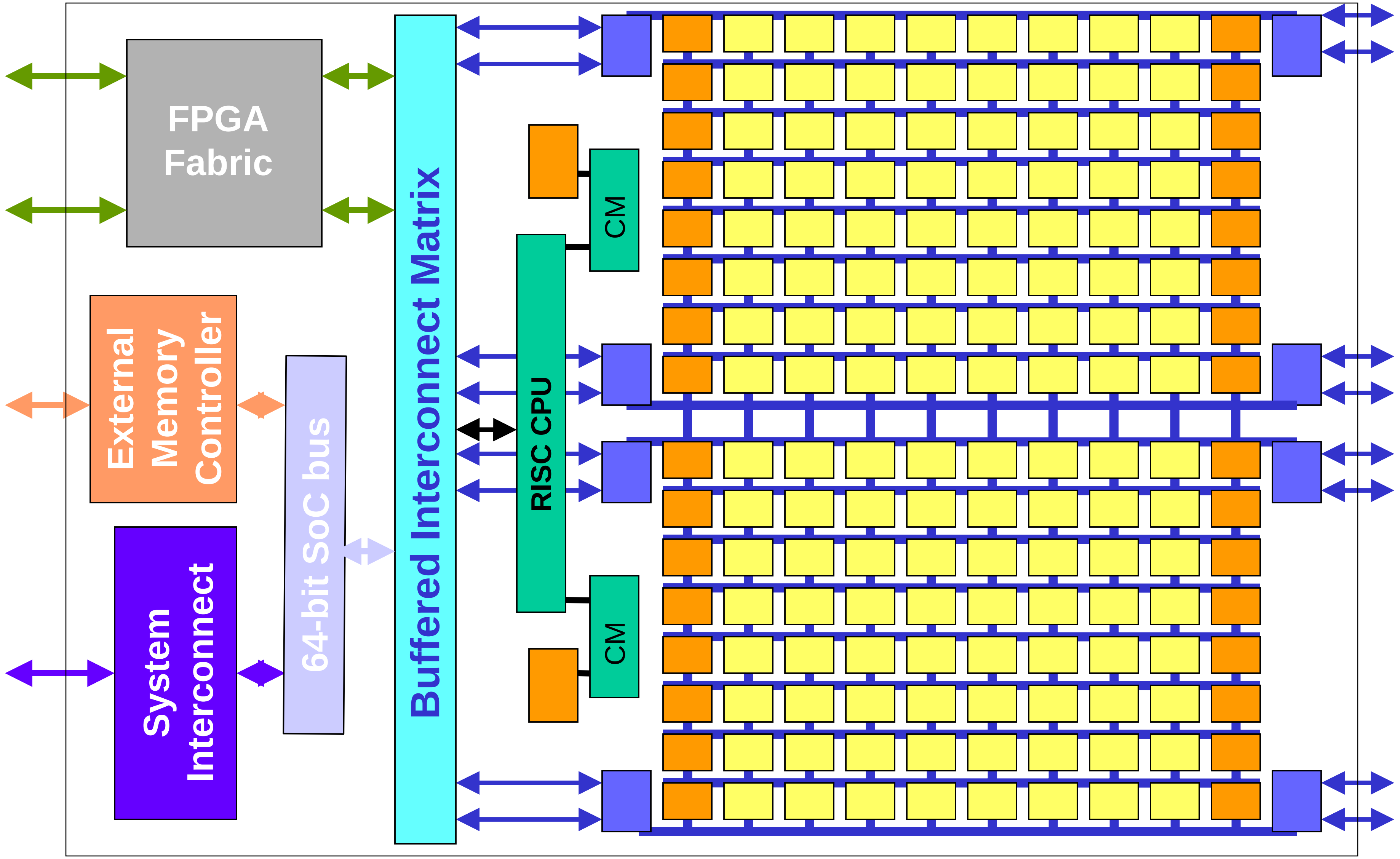
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**13**	

The eXtreme Adaptive DSP Solution to Sensor Data Processing

Martin Vorbach*, PACT XPP*

Leo Mirkin, *SKY Computers, Inc.*

Example of eXtreme DSP Architecture - QuickMIPS-XPP



Direct Compilation from C Language onto Computing Grid

- eXtreme DSP processor combines RISC CPU with a scalable DSP coprocessor made from a regular grid of Processing Array Elements (PAE) and supported by FPGA-controlled I/O interface

- PAE grid is made from 16-32 bit fixed or floating point MAC-ALU with streaming I/O cells delivering external data and RAM cells holding constants and intermediate results

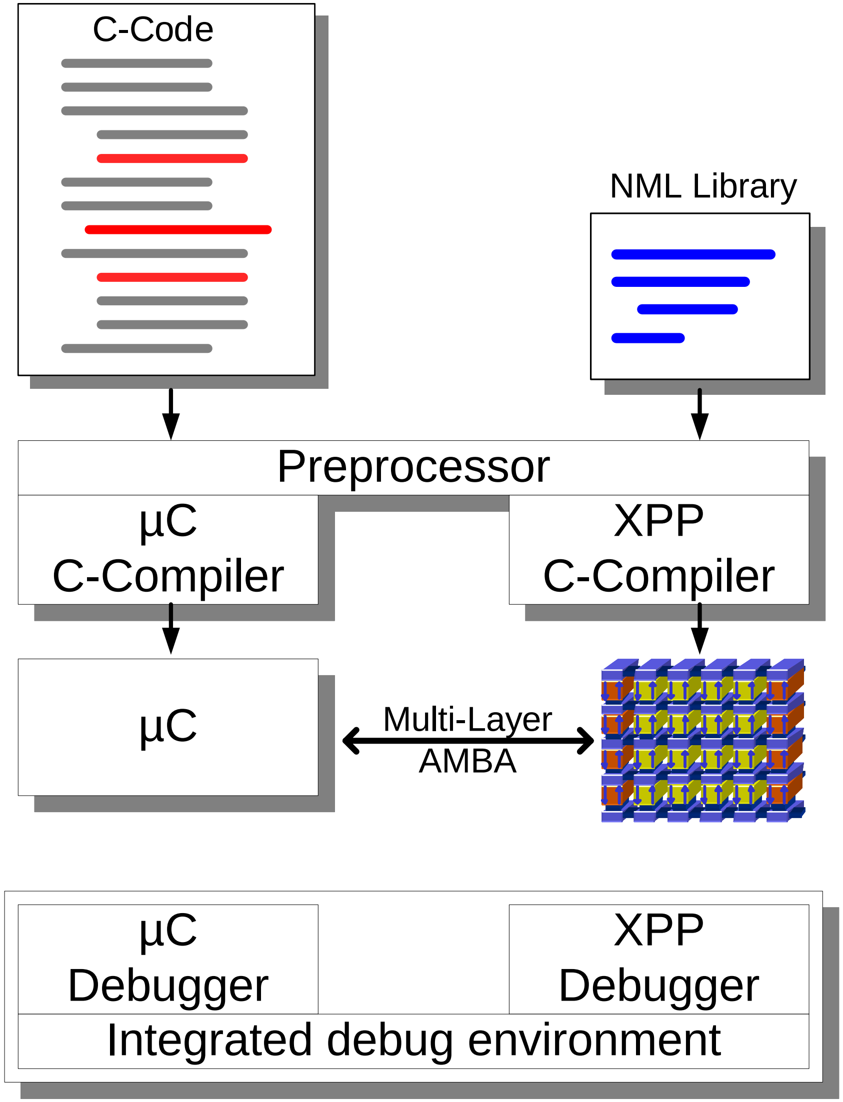
- Grid regularity and PAE control simplicity allows DSP programming by unrolling legacy C-code into the space of computing grid

- All DSP processing carried by scalable XPP co-processor with non-critical scalar and glue code running in RISC CPU

Key Elements of the eXtreme DSP Architecture

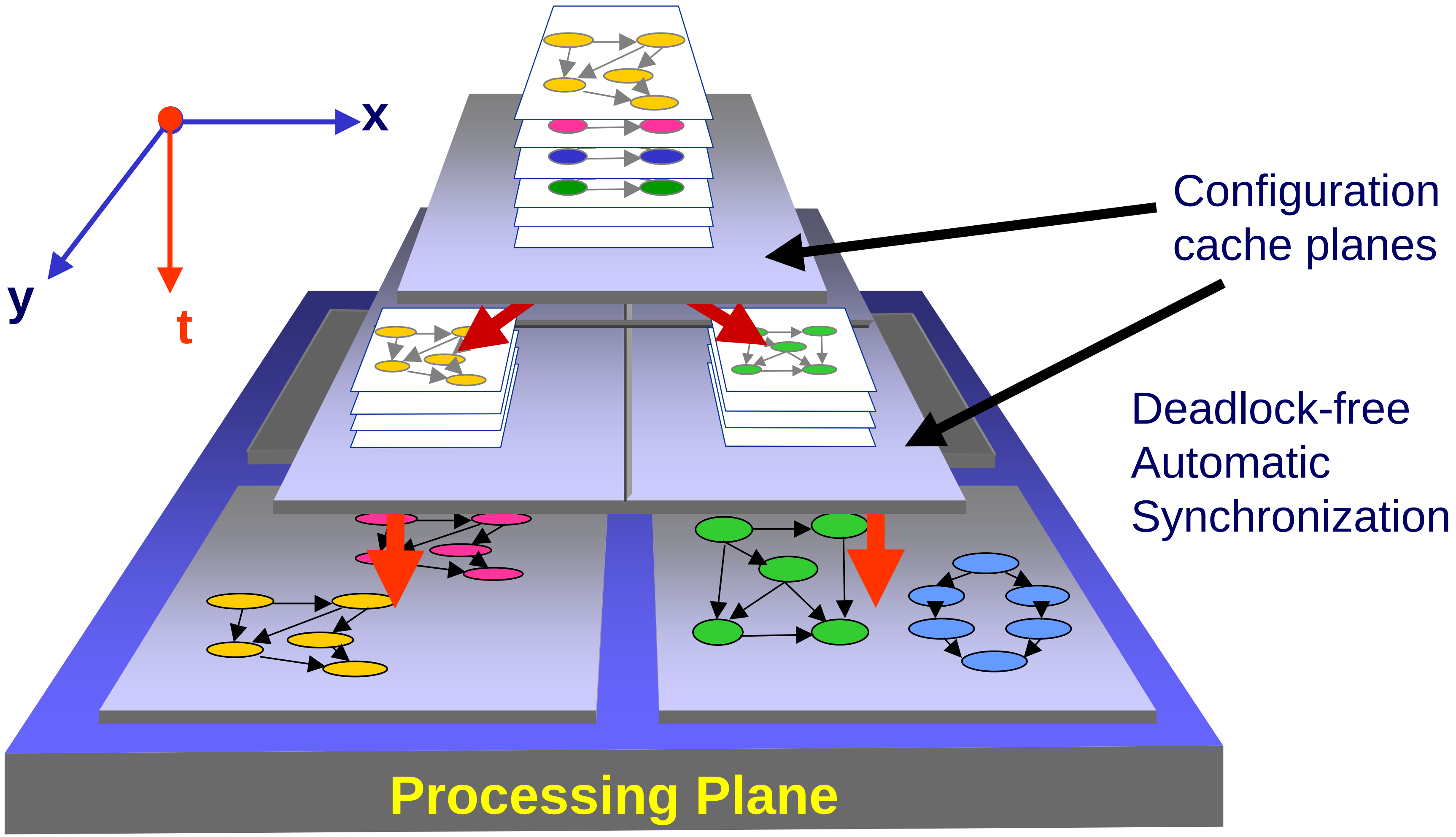
- PAE grid is tied together by a data packet carrying high-bandwidth interconnect
- Configuration network links all PAEs and operation of each PAE can be independently re-programmed in one cycle
- Automatic data flow synchronization makes PAE operations deadlock-free
- PAE operations are data driven and *don't consume power in the absence of input data*

Integrated C-based DSP Programming for QuickMIPS-XPP

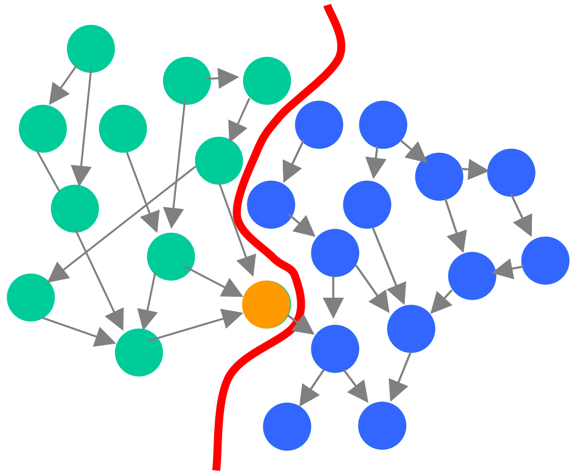


- eXtreme DSP tools integrated in host processor C-toolchain
- One source code for XPP and µC
  - Code exchanged by
    - -Source code annotation
    - -Library subroutine calls
- Automatic insertion of interface routines for µC and XPP intercommunication
- Fully integrated debug environment

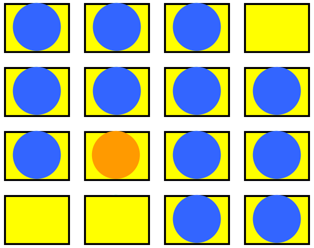
DSP Fabric Configuration and Deadlock-Free Synchronization



Fitting Large Algorithm to XPP Grid Using Instant PAE Reprogramming

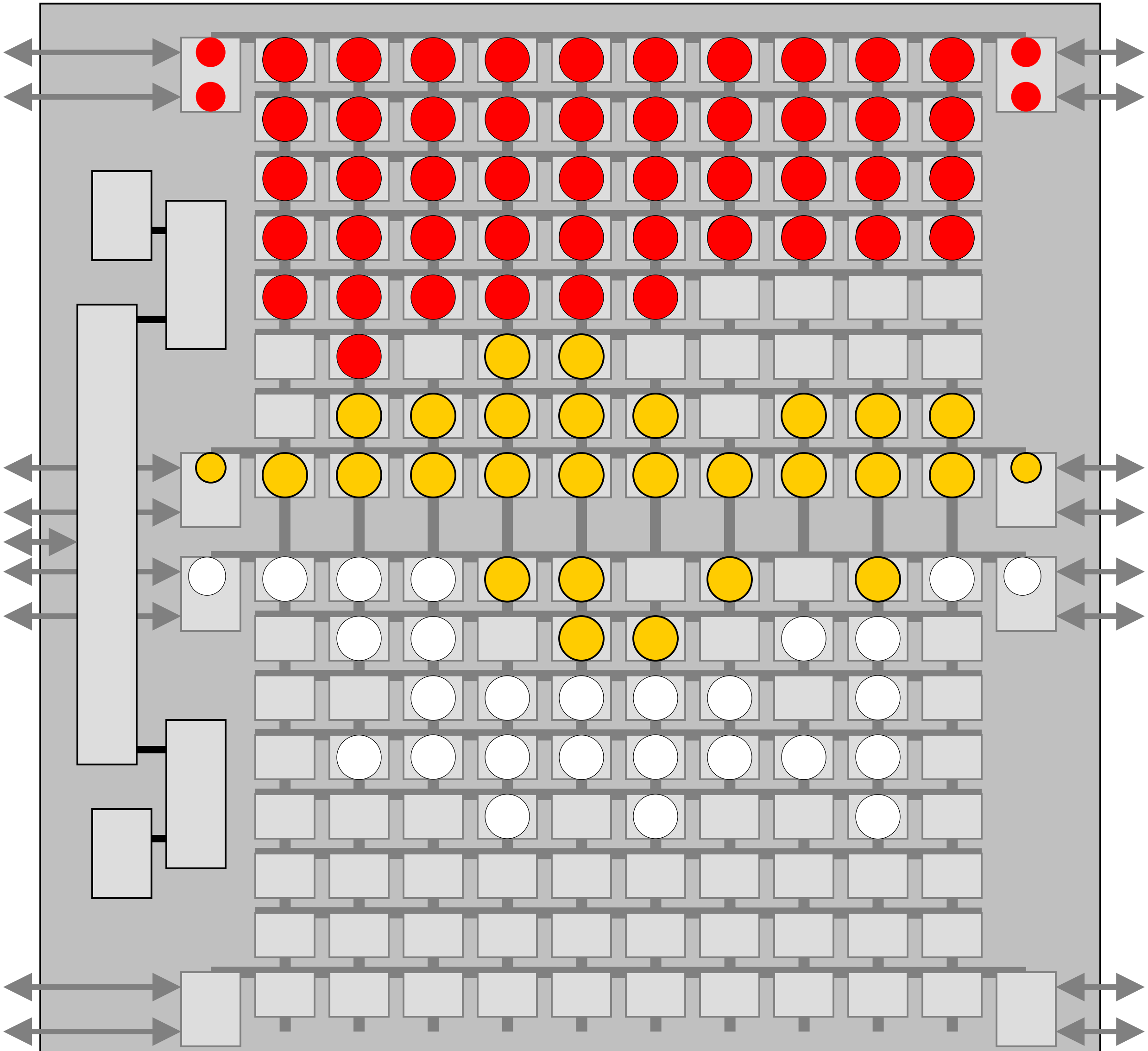


- If large flow graph does not fit into PAE grid
- First, locate a good separation point, partition graph into parts **1** and **2** using shared PAEs or Memory as a destination
- Program partition 1 into XPP grid



- After calculation, remove partition 1
- Data is still available in shared ALU-PAE or in RAM-PAE
- Re-program XPP grid and compute part 2

Partial One-Cycle Reconfiguration Supports Adaptive Processing



Task 2

Task 3

Task 4

Task 5 is partially configured

waiting for Task 2 resources to become free

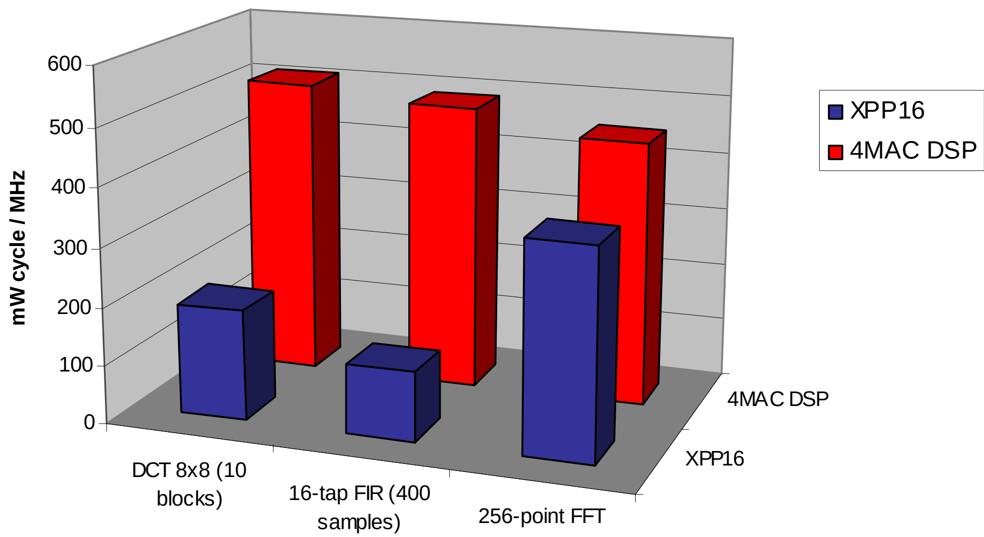
Configuration is completed

PACT

Power-Efficient DSP Computing

QUICKLOGIC®

Algorithm		XPP16	4xMAC DSP
FFT 256-points	*mW/ MHz*	360	453
	cycles	1,200	1,619
MPEG Video 2D DCT (8x8)	*mW/ MHz*	19	51
	cycles	64	181
Real 16 Tap FIR Filter (40 Samples)	*mW/ MHz*	12	49
	cycles	40	176



- **XPP trades clock frequency for high spacial parallelism**
- **Saves power by dramatically reducing need for**
  - **opcode fetch and decode**
  - **temporary data transfer to register/cache/memory**

Key Engineering Advantages of eXtreme DSP

- Gaining performance by trading silicon space for higher clock

- Familiar C-language programming model for all grid sizes dramatically speeds up software development and verification

- Getting ASIC/FPGA-level optimal DSP performance combined with full or partial on the fly re-programming

- Elimination of the unnecessary gate switching delivers power efficient DSP computing

- Processor versions with different PAE grid sizes offer wide range of DSP performances with identical programming model

Addressing Critical Needs of COTS DSP Programs

eXtreme DSP architecture:

- Provides significant increases in DSP performance while lowering power consumption
- Drastically speeds-up design and upgrade cycles and simplifies technology upgrades for legacy products
- Makes DSP software portable between product generations
- Assures long-term economical viability of design by riding on future semiconductor density increases (the Moore's Law)

headroom to explore new programming paradigms that range from a visual data-flow framework to new variations of a parallelizing compiler.

The eXtreme programming tool transforms a source code into a data-flow and control-flow graph and creates a configuration that maps nodal operations directly onto the array for parallel execution. It allows for tunable variations that span from the traditional Von Neumann architecture to a highly power-efficient data-driven parallel processing of multiple data-flow threads. These variations can be enacted differentially and without any interruptions by a unique process of self-reconfiguration.

The result shows unprecedented processing power and data bandwidth in a power-thrifty package. The peak performance of the first commercial device running at a very modest 50MHz achieved 19.2GOPS/sec, and the peak I/O bandwidth of 3 GB/sec.

The eXtreme processor is:

More powerful than DSP, with lower power dissipation and better programmability

Cheaper than an equivalent FGPA, with lower power dissipation, and much easier to program and maintain

Cheaper and faster to develop than an ASIC

Much more powerful than general-purpose uP at a much lower power consumption and a much higher data bandwidth without additional chips

Linear-scale performance gains can be achieved by an increase in the size of the XPP array **without perturbing the architecture**. Gaining more speed with progressively smaller geometries, the eXtreme technology takes advantage of the XPP array expansion while simultaneously keeping in check the clock frequency and minimizing total power consumption. Its natural scalability also allows the eXtreme to take full advantage of future increases in the device densities predicted by Moore's Law. Longevity of a COTS DSP solution engineered with the eXtreme processors will be assured by the timely and software-compatible performance upgrades.

From the commercial viability point of view, architecture of the eXtreme processors is **generic enough to satisfy many niche signal-processing markets,** with the total market volume sufficient to recoup engineering and manufacturing costs. The optimal performance of eXtreme processors results from their ability to match the structure of data-flow-driven algorithms. Such good fit will assure their acceptance and longevity in the COTS DSP market.

Compared to an FPGA, the eXtreme architecture provides the **right design granularity with the right configurability** to allow a direct code generation from the traditional high-level languages to be optimized for a given set of tasks. It also guarantees a much **simpler and faster design verification**. The differential and wave reconfiguration available in the eXtreme architecture is impossible for an FPGA all-or-nothing reconfiguration process.

The relatively coarse granularity of the eXtreme design allows for naturally adaptable power management: the XPP-array computing elements are idle in the absence of data and minimize their power consumption. Overbuilding a DSP system with more eXtreme processors than application requires, automatically "right-sizes" power consumption by idling unused parts of the computational array.

The eXtreme XPP arithmetic units can be of any width from 16 to 32 bits (and even more) with the floating-point included as a design option.

We will discuss the XPP implementation of typical DSP algorithms: FIR and FFT in comparison to Motorola and TI DSP.